

---

## System- und Schnittstellenentwurf

---

Team: sysopr12

Autoren: Thomas Merkel (19868)  
Janik Lemcke (19865)  
Arkadiusz Rawa (19893)

Letzte Änderung: Dienstag, 2. November 2010

Anmerkung: Überarbeitete Version nach Teambesprechung am Dienstag, 2. November 2010

# Inhaltsverzeichnis

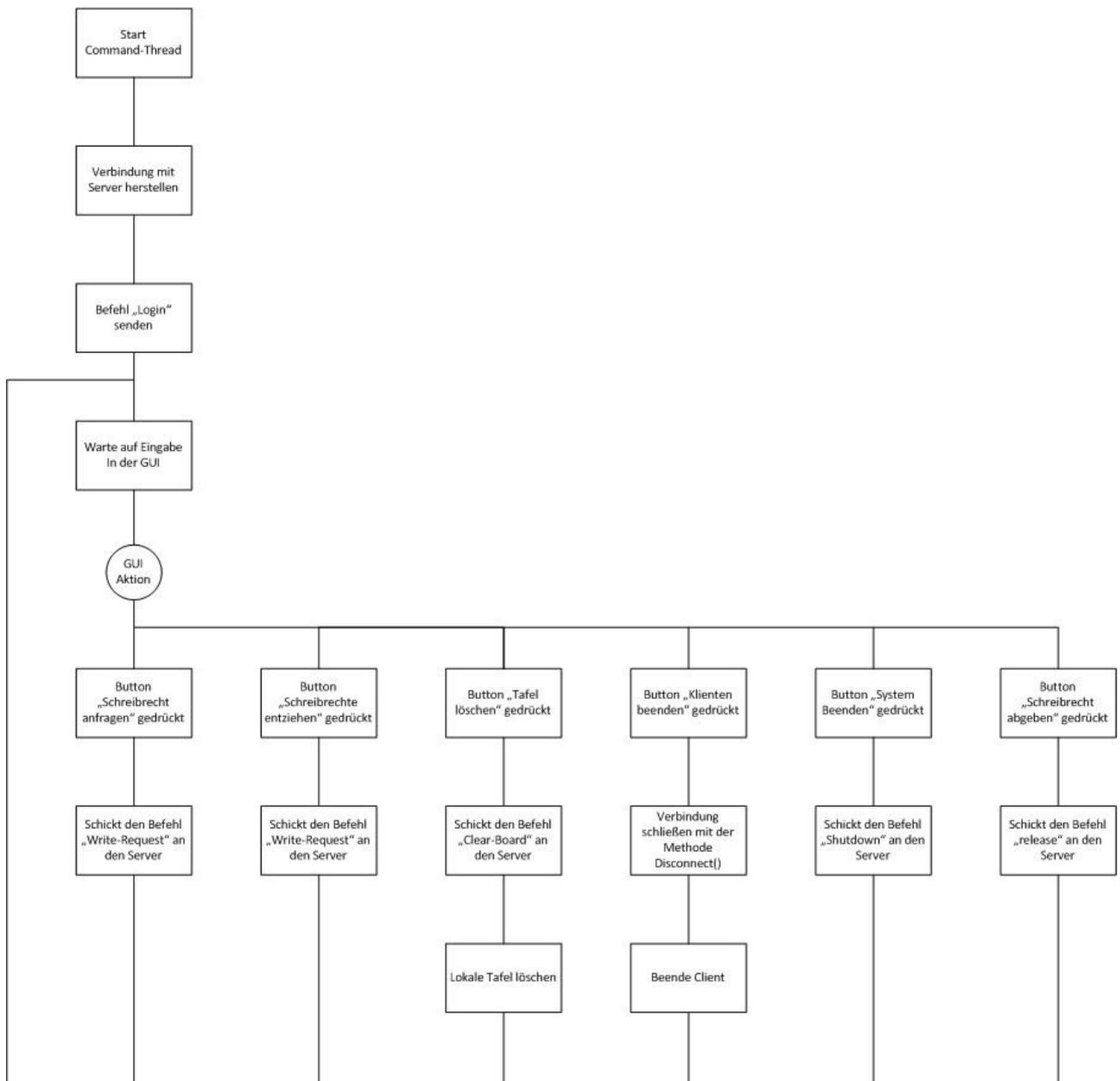
1 Client.....	3
1.1 Command Thread.....	3
1.1.1 Ablaufdiagramm.....	3
1.2 Listener Thread.....	4
1.2.1 Ablaufdiagramm.....	4
1.3 Live Agent.....	5
1.3.1 Ablaufdiagramm.....	5
2 Server.....	6
2.1 Login Thread.....	6
2.1.1 Ablaufdiagramm.....	6
2.2 Client List.....	7
2.3 Client Thread.....	7
2.3.1 Ablaufdiagramm.....	8
2.4 Broadcast Agent.....	9
2.4.1 Ablaufdiagramm.....	10
2.5 Archivierer.....	11
2.5.1 Ablaufdiagramm.....	11
2.6 Logger.....	12
2.6.1 Ablaufdiagramm.....	12

# 1 Client

## 1.1 Command Thread

Der Command Thread reagiert auf die Aktionen, Benutzung der Tastatur, des Benutzers und teilt diese dem Server mit. Dies geschieht über eine eindeutige Socket Verbindung.

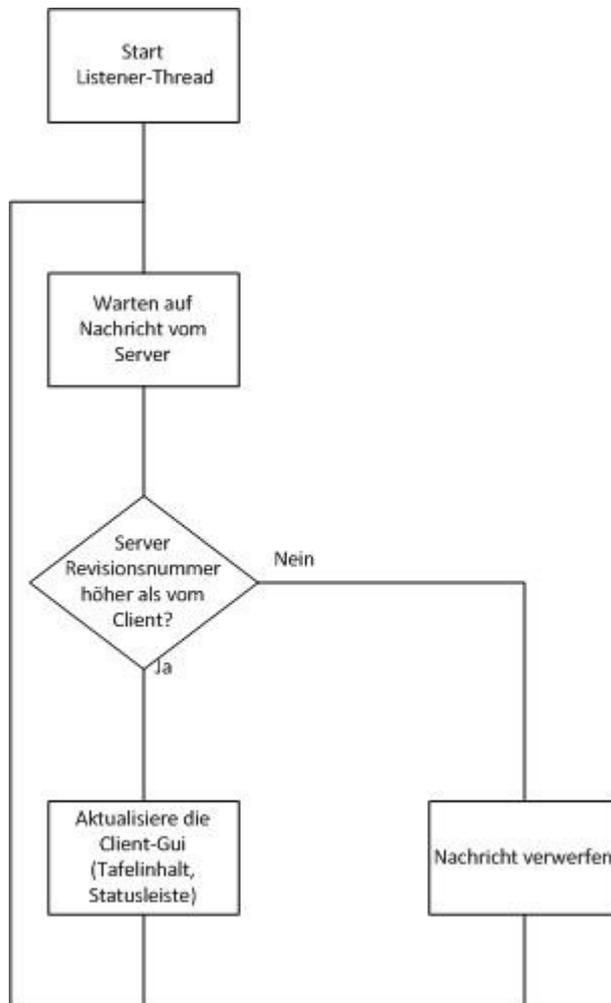
### 1.1.1 Ablaufdiagramm



## 1.2 Listener Thread

Der Server benachrichtigt den Listener Thread des Clients, dass es Tafeländerungen gab. Der Listener vergleicht die Revisionsnummern des Servers und die eigene, wenn die Revisionsnummer höher ist als die Eigene dann wird die Tafel des Clients aktualisiert. Ansonsten wird diese Nachricht verworfen.

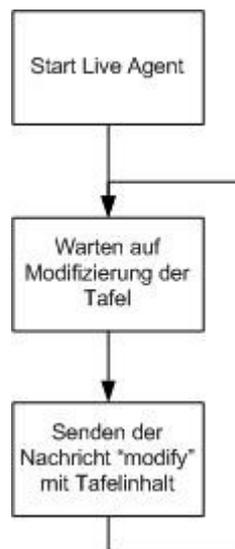
### 1.2.1 Ablaufdiagramm



## 1.3 Live Agent

Sobald der Client Schreibrechte besitzt wird dieser Thread aktiv, denn er sendet eine „modify“ Nachricht an den Server sobald es Änderungen an der Tafel gab. Der Server leitet diese an den Broadcast Agenten weiter.

### 1.3.1 Ablaufdiagramm



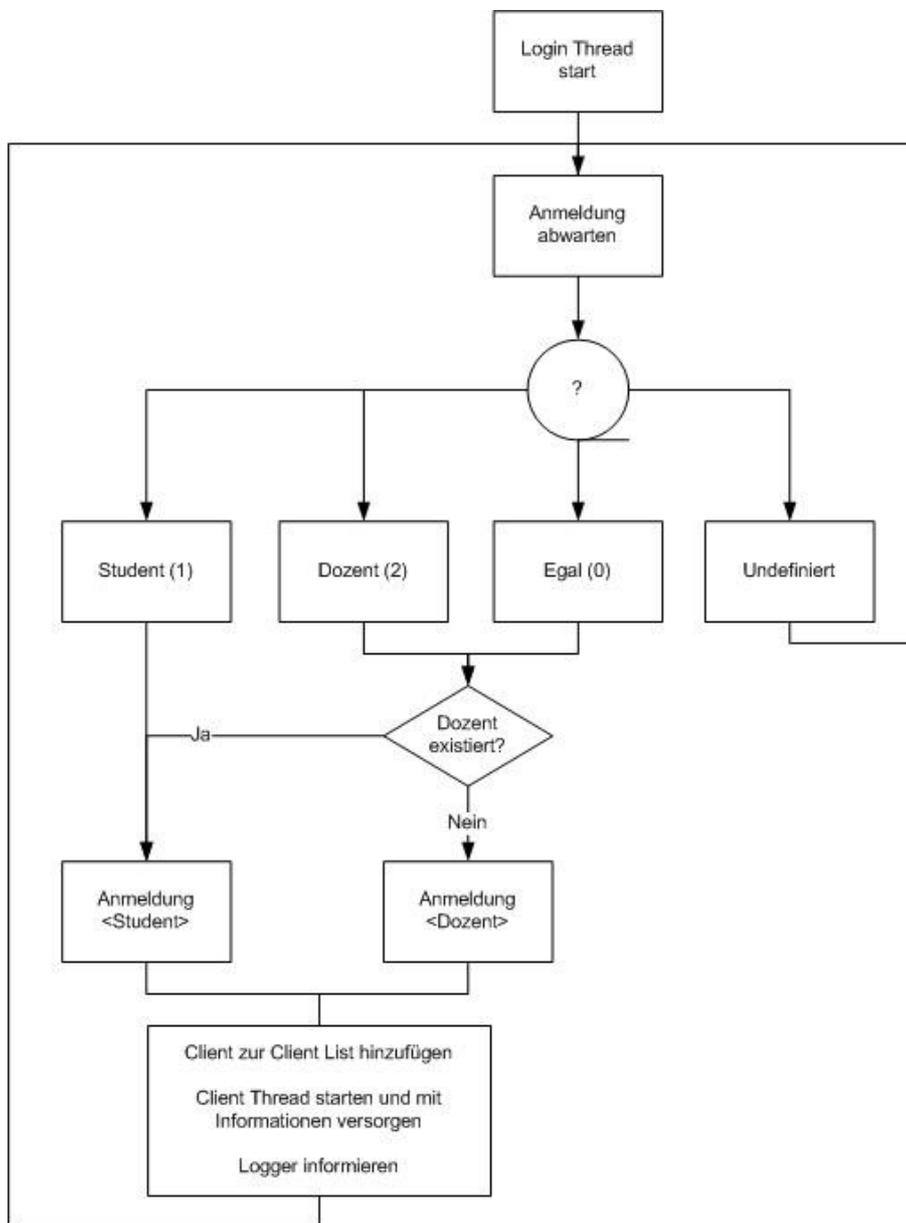
## 2 Server

### 2.1 Login Thread

Dieser Thread wird beim Starten des Servers geforked, das heißt automatisch mit gestartet.

Sobald eine Anmeldung am Server erfolgreich ist, werden die nötigen Informationen an die Client Liste übertragen. Hierbei wird auch ein separater Client Thread pro Client gestartet, dieser wartet nun auf Befehle.

#### 2.1.1 Ablaufdiagramm



## **2.2 Client List**

Speicherung und Verwaltung wichtiger Informationen die für die Client Kommunikation nötig sind.

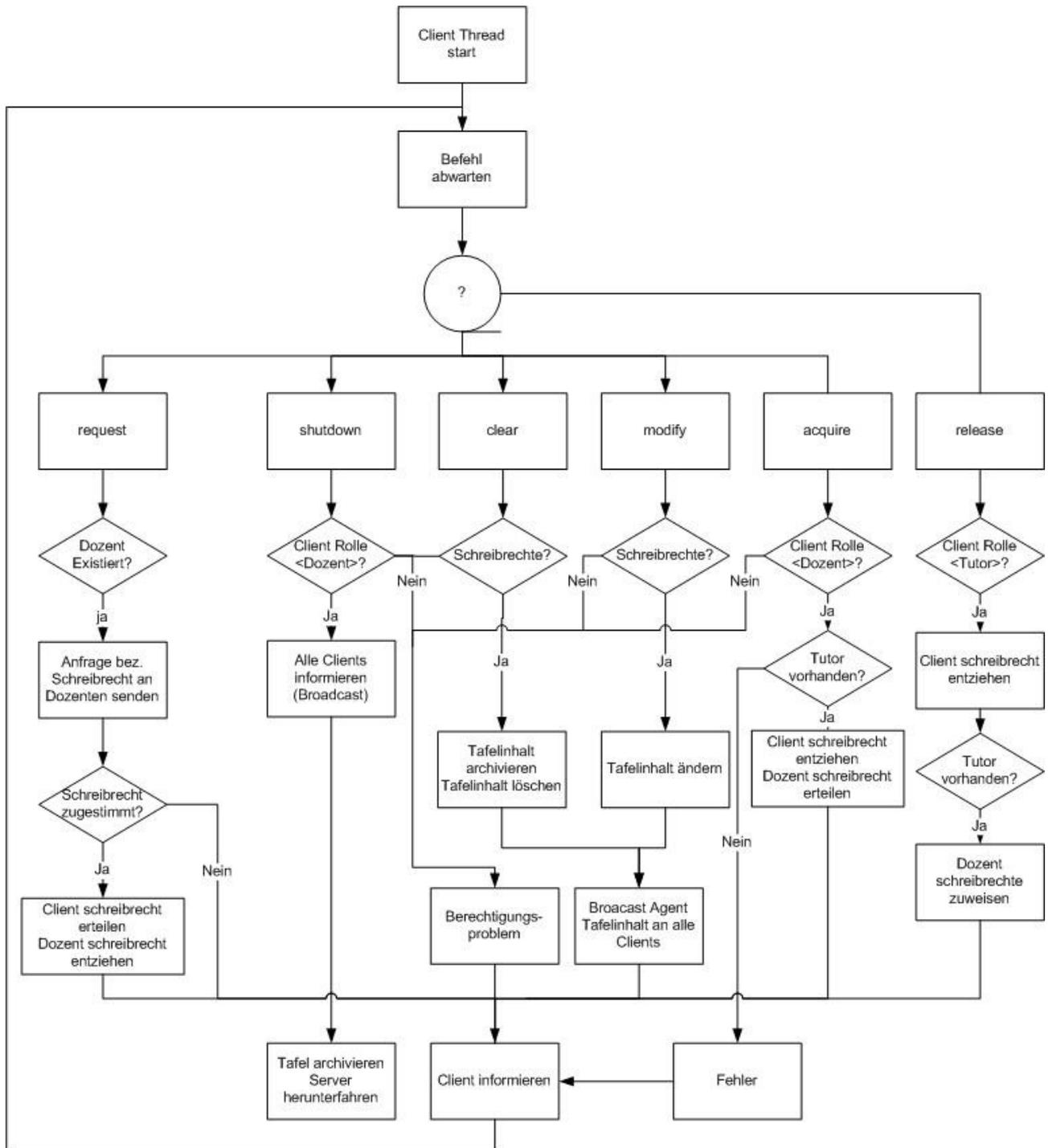
- Client Socket
- Client ID
- Client Name
- Client Rolle

## **2.3 Client Thread**

Es wird bei jedem erfolgreichen Login ein neuer Client Thread gestartet.

Dieser wartet auf Kommandos vom Client und führt die entsprechenden Aktionen durch. Nach dem diese ausgeführt wurden, schickt der Client Thread dem Client eine Ergebnis Nachricht zurück.

### 2.3.1 Ablaufdiagramm



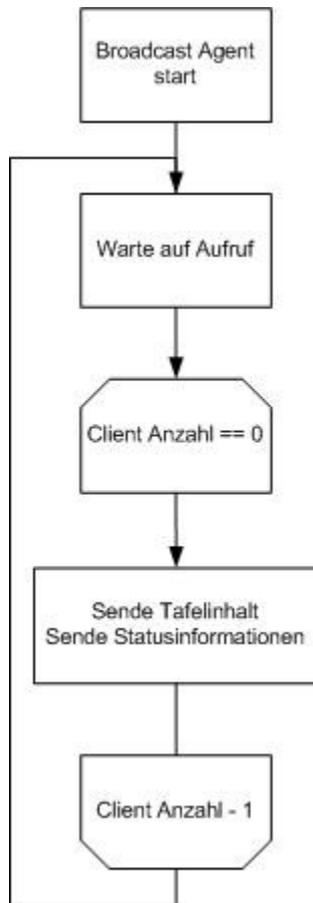
## 2.4 Broadcast Agent

Es wird auf Befehle vom Client Thread gewartet, sobald ein entsprechender Befehl eintrifft wird diese Information an alle Clients verteilt. Der Broadcast Agent muss bei allen Befehlen bis auf *acquire* reagieren.

Folgende Daten werden vom Broadcast Agent übertragen:

- Tafelinhalt
- Statusleiste
  - Client ID
  - Schreibrechte (0 = keine Schreibrechte, 1 = Schreibrechte)
  - Anzahl Dozenten
  - Anzahl Tutoren
  - Anzahl Stufenente

### 2.4.1 Ablaufdiagramm

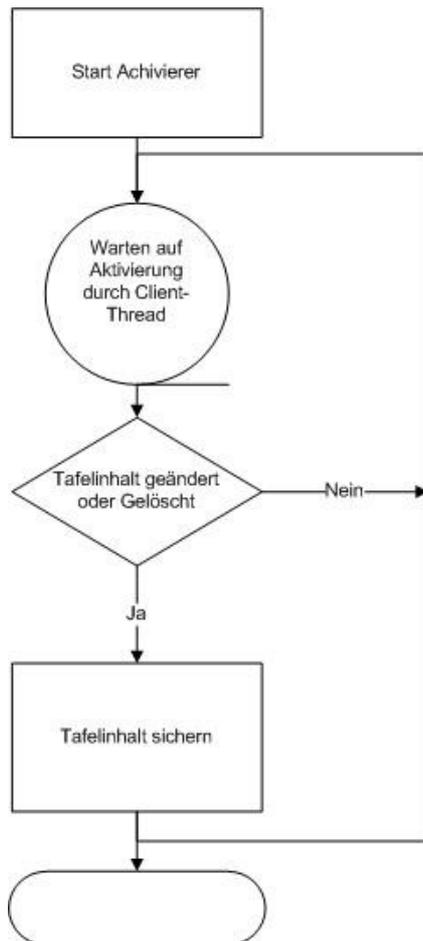


## 2.5 Archivierer

Der Archivierer wird von dem Server aus gestartet und auch beendet.

Dieser sorgt dafür, dass beim empfangen des Befehls „clear“ der alte Tafelinhalt in eine Archivdatei gespeichert wird und somit alle Änderungen an der Tafel nachvollzogen werden können.

### 2.5.1 Ablaufdiagramm



## 2.6 Logger

Der Logger wird von dem Server aus gestartet und auch beendet.

Wenn sich ein neuer Client am Server anmeldet, ein angemeldeter Client einen Befehl an den Server schickt oder der Broadcast-Agent den aktuellen Tafelinhalt an alle Clients sendet, wird eine Protokollnachricht an die Message-Queue des Loggers gesendet, welcher daraufhin diese Nachricht in eine Logdatei schreibt.

### 2.6.1 Ablaufdiagramm

